

Mundo do *Wumpus*

João Vitor, Millas Násser, Paulo Tobias, Welton Santos

Novembro 2017

1 Introdução

A proposta deste trabalho é apresentar uma solução para um problema de IA conhecido como "*Mundo do Wumpus*".

Ele consiste em buscar um tesouro em um ambiente completamente desconhecido, onde há um mostro chamado *Wumpus* que pode matá-lo.

Além disso em determinadas posições do mapa existem buracos que causam uma morte instantânea para o agente.

Portanto o objetivo é buscar o tesouro evitando todos os contratempos utilizando conceitos de lógica para fazer suposições sobre posições não visitadas.

2 Modelo da estrutura

Para fazer o modelo do mundo do agente foi utilizado uma matriz que mantém os estados em cada posição.

Para representar cada estado é utilizado uma 8-tupla com cada termo variando entre 0 e 1 onde (BRISA, FEDOR, POÇO, WUMPUS, RELUZENTE, JOGADOR, CONHECIDO, VISITADO)

- BRISA, indica que há um poço por perto.
- FEDOR, indica que o monstro *Wumpus* está próximo à aquela posição.
- POÇO, lugar exato do poço.
- WUMPUS, local exato do monstro.
- RELUZENTE, local que possui o tesouro.
- JOGADOR, posição que o agente está no mapa.
- CONHECIDO, se o estado é conhecido por alguma forma de inferência anterior.
- VISITADO, se o estado já foi visitado pelo agente.

Desta forma se o estado corrente possui as percepções FEDOR, BRISA e RELUZENTE a 8-tupla que a representa é (1,1,0,0,1,1,0,1), isso pois ele sentiu as duas percepções, aquela posição contém o jogador e como que ele está presente também indica que ela foi visitada.

Mas pode acontecer de posições que são conhecidas em algum momento, estas carregam a percepção de conhecido e não possuem o jogador. Normalmente são inferências feitas sobre esta posição.

3 O algoritmo

Para resolver o paradigma foi utilizado o conceito de recursão, portanto quando o agente chega a um fim caminho ele volta para o pai.

A função recebe dois parâmetros principais: Uma lista de posições já visitadas no estado atual da recursão e uma variável *booleana* indicando se a execução corrente pode tomar ações drásticas, i.e., atirar uma flecha ou andar em direção a um poço.

O primeiro passo é verificar o que existe na posição atual do agente. Se for um poço ou *Wumpus*, então o jogo é encerrado. Se for ouro, então a ação AGARRAR é enviada e o jogo também termina.

Se nenhum dos três estados estiver na posição do agente, então ele verifica o mundo conhecido para inferir onde está o *Wumpus* e os poços.

Após isso, as três listas são preenchidas usando as posições adjacentes ao agente.

Enquanto houver alguma lista não-vazia: Se existe uma posição livre na lista, então o agente se moverá para ela. Caso contrário, se a lista de *Wumpus* não estiver vazia, a função tem dois caminhos: Se a variável ultimo recebida por parâmetro for verdadeira, então o agente irá atirar no *Wumpus* ou se moverá em direção a ele. Se ultimo for falsa, então o agente voltará para a posição anterior e retornará para a função que a chamou que encontrou um *Wumpus* no caminho. Por fim, se a lista de *Wumpus* também estiver vazia, então a função também tem dois caminhos: Se a variável ultimo recebida por parâmetro for verdadeira, então o agente escolherá um poço da lista aleatoriamente e se moverá em direção a ele. Se ultimo for false, então o agente voltará para a posição anterior e retornará para a função que a chamou que encontrou um poço no caminho.

Após a ação tomada, e recursão é chamada. Caso o agente tenha andado, ele se insere na lista de posições visitadas. O valor da variável ultimo depende dos seguintes fatores: Se a função recebeu ultimo como falso, então ela somente poderá passar falso adiante. Se ela recebeu ultimo como verdadeiro, então ela passará ultimo como verdadeiro caso exista apenas uma posição livre e a lista de *Wumpus* esteja vazia.

O retorno da chamada recursiva é então analisado. Se retornar que encontrou um *Wumpus*, então essa posição é adicionada na lista de *Wumpus*. Se retornar que encontrou um poço, então essa posição é adicionada na lista de poço.

4 Análise de resultados

4.1 Casos especiais

Segue abaixo algumas situações que requereram mais ações externas para a IA conseguir prosseguir ou inferir certas informações.

RF		B	
W	FB	PB	B
FB	PB	P	B
J	B	B	

Figura 1: Problema 1: Tentativa de liberar uma casa matando Wumpus

No mapa acima, o jogador inicialmente ficará em uma situação em que ele não consegue avançar com segurança para nenhuma casa adjacente, seja por causa de brisas resultando em possíveis poços ou fedores resultando nas localizações possíveis do *Wumpus*. Nesse caso, a tentativa adotada para tentar controlar o dilema é atirar a flecha na direção que o jogador supôs que o Wumpus está (caso tenha conseguido concluir sua hipótese, se não, a direção da flecha será aleatória entre as duas casas possíveis).

B	P		
	B	P	
		B	P
		BJ	P

Figura 2: Problema 1: Mundo conhecido do jogador

Se a flecha matou o monstro, o jogador elimina pelo menos dois fedores, uma ou mais possibilidade de Wumpus e talvez consiga liberar uma casa no mapa, caso contrário, o jogador pode chegar a eliminar uma possibilidade falsa de Wumpus. Entretanto, mesmo que o Wumpus morra, caso o mesmo esteja em um buraco, esta tentativa praticamente não contorna o problema em questão, pois o poço naquela posição continuará emitindo sua brisa.

B	RB		
P	PB	FB	B
B	FB	WB	FP
J		F	B

Figura 3: Problema 2: Jogador cercado de poços

Além deste dilema, ainda existe a situação em que o jogador "se cerca" com possibilidades de poços, sem conseguir eliminar nenhuma possibilidade falsa. Esse pode ser considerado um dos casos mais difíceis possíveis, pois basicamente não é possível inferir nenhuma localização de poço, restando apenas escolher uma casa aleatória e tentar andar nela sem morrer. Esta é a única escolha que a IA tem.

P	WP		
B	FB	W	
		FJ	

Figura 4: Problema 2: Mundo conhecido do jogador

B	P	BF	W
	B	P	BFR
		B	B
J		B	P

Figura 5: Problema 3: Três possibilidades de Wumpus

Mais uma situação que necessitou de mais verificações para ser resolvida foi a que o jogador geraria três possibilidades de uso ao percorrer em volta da localização verdadeira do Wumpus. No exemplo abaixo, além das duas casas que o jogador já supõe que o Wumpus está, antigamente ele iria gerar uma terceira possibilidade errônea na casa [3,3] graças ao fedor presente nesta casa.

B	P		
	B	P	
		B	P
		BJ	P

Figura 6: Problema 3: Mundo conhecido do jogador

O Wumpus só pode estar em apenas uma de duas possibilidades geradas à partir de um fedor. Se o jogador encontra fedor em outra casa, então o Wumpus está na casa em que foi gerada duas vezes a possibilidade do Wumpus estar nela. Isso agora é inferido pelo jogador, poupando pelo menos duas casas a mais para o jogador não se preocupar pelo menos com o monstro.

Entretanto para 100 casos gerados randomicamente, houve 97 acertos e 3 resultaram em fim de jogo. Considerando que o algoritmo só pode falhar quando

realmente o jogador tiver que escolher uma casa aleatória para prosseguir no mapa e errar, os resultados foram muito bons.

5 Conclusão

Em suma, a IA do jogador mostra-se bastante satisfatória. Quando não existem casos duvidosos para o jogador tentar prosseguir, o algoritmo consegue explorar o mapa com eficiência. Mesmo quando existe a dúvida para onde andar por que todas as possibilidades anteriores falharam, ainda existem certas tentativas de inferência que talvez ajude no próximo passo a seguir na mapa. Quando todas as tentativas se esgotaram, então a única forma lógica de continuar o jogo é contar com a sorte ao andar em uma casa apontada como poço.